



TITLE:

ある非有界無限区間積分の高速高精度計算 (科学技術計算における理論と応用の新展開)

AUTHOR(S):

大浦, 拓哉

CITATION:

大浦, 拓哉. ある非有界無限区間積分の高速高精度計算 (科学技術計算における理論と応用の新展開). 数理解析研究所講究録 2012, 1791: 141-150

ISSUE DATE:

2012-04

URL:

<http://hdl.handle.net/2433/172830>

RIGHT:

ある非有界無限区間積分の高速高精度計算

京都大学数理解析研究所 大浦拓哉 (Takuya OOURA)

Research Institute for Mathematical Sciences,
Kyoto University

1 はじめに

解析概論 [1] の練習問題 (p.141) に出てくる積分

$$I = \int_0^{\infty} \frac{x}{1 + x^6 \sin^2 x} dx \quad (1)$$

は, 20 世紀初頭に E. Goursat[2], G. H. Hardy[3] らによって収束することが確認されたものである. 本稿ではこれを, Goursat-Hardy 積分 (GH 積分) と呼ぶ. この積分に関して, 1984 年京都大学数理解析研究所の研究集会で戸田英雄は, 収束性の議論にとどまらず, 数値計算の問題として捉えることを提起した. 翌年, 二宮市三は積分算法と加速法を駆使して約 21 桁の計算を行い [4], 2009 年, 秦野甯世, 二宮市三, 杉浦洋, 長谷川武光らは, 8 倍精度と周回積分で約 73 桁を計算した [5].

本稿は, GH 積分の計算法を演算量の削減という視点からさらに改良し, 徹底した高速高精度計算を行うことを目的とする. さらに, 求める桁数と必要な演算量との関係についての考察を行う.

具体的には, GH 積分を佐藤超関数とみなして, Hilbert 変換によって変形し, 穏やかな関数の積分に変換する. その上で, 二重指数関数型数値積分公式 (DE 公式)[6] を適用し, 高速高精度計算を可能にするというものである.

また, 更なる高速高精度計算を可能にするため, この積分に特化した DE 公式を作成する. その DE 公式は, N 桁の積分値を得るための関数計算回数が $O(N)$ となる超収束する DE 公式である. 最後に 100 万桁以上の実計算を行い, 桁数と計算量の関係について考察する.

2 GH 積分の計算困難性

GH 積分 (1) の被積分関数を図 1 に示す. この関数の積分値を, 一般的な積分公式で N 桁求めようとする場合を考える. 粗い解析で, この関数は $x \approx k\pi$ ($k = 1, 2, 3, \dots$) の付近に高さ $O(k)$, 半値幅 $O(1/k^3)$ のピークがあり, 区間 $[0, k\pi]$ の積分値は $O(1/k)$ の誤差で収束する. したがって, N 桁求めるためには区間 $[0, 10^N \pi]$ の計算が必要である. さらに, ピークの幅よりも積分公式の刻み幅を十分小さくしなければならず, 例えば端点がピークとなる複合 Gauss-Legendre 公式を用いる場合は, 分点数が $O(Nk^{3/2})$ となる積分計算を $k = 1, 2, \dots, 10^N$ に関して行わなければならない. したがって, この場合の N 桁求めるための関数計算回数は $O(N \cdot 10^{5N/2})$ となり, 桁数に対して指数関数オーダーの計算量が必要となる.

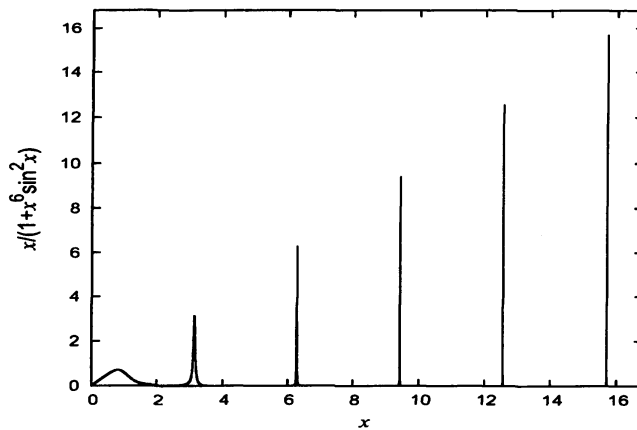


図 1: GH 積分の被積分関数

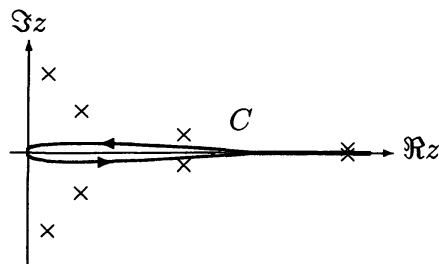
3 Hilbert 変換による変形

まず, 積分 (1) の被積分関数を複素関数として解析すると, $0.3 \pm 0.9i$, $0.9 \pm 0.4i$ 付近と, $\pi k \pm i/(\pi k)^3$ ($k = 1, 2, 3, \dots$) 付近に極がある [5]. それを踏まえ (1) を複素積分へ置き換える.

$$I = \frac{1}{2\pi i} \int_C \varphi(z) dz,$$

$$\varphi(z) = \begin{cases} \frac{-\pi iz}{1 + z^6 \sin^2 z}, & \Im z > 0 \\ \frac{+\pi iz}{1 + z^6 \sin^2 z}, & \Im z < 0 \end{cases}$$

積分路は, 図 2 に示すように, 極の内側で実軸近傍の境界を這わせるようにとる.

図 2: 積分路 C

次に, この極を消し去るために, (1) の被積分関数を少し変形した関数の Hilbert 変換

$$\psi(x) = \text{pv} \int_{-\infty}^{\infty} \frac{x}{1 + x^6 \sin^2 y} \cdot \frac{dy}{x - y}$$

を考える. ここで, pv は主値積分を意味する. この変換は解析的に計算できて,

$$\psi(z) = \frac{\pi z^6 \sin z \cos z}{\sqrt{1 + z^6}} \frac{z}{1 + z^6 \sin^2 z}$$

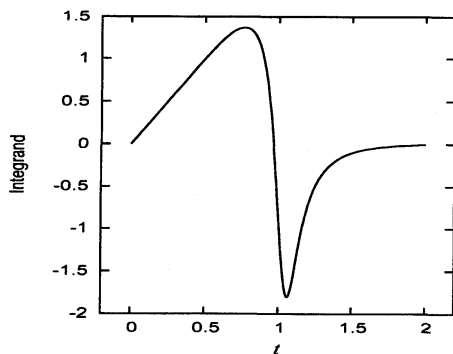
となる。少しの解析で、 $\psi(z)$ と $\varphi(z)$ の分岐点を除く極は、すべて同じ位置で $\psi(z)$ と $\varphi(z)$ の留数の符号が逆になることがわかる。また、 $\psi(z)$ は実軸近傍で正則なので、 C 上の積分値はゼロである。そこでこれらの性質を利用して、複素積分を

$$I = \frac{1}{2\pi i} \int_C (\varphi(z) + \psi(z)) dz$$

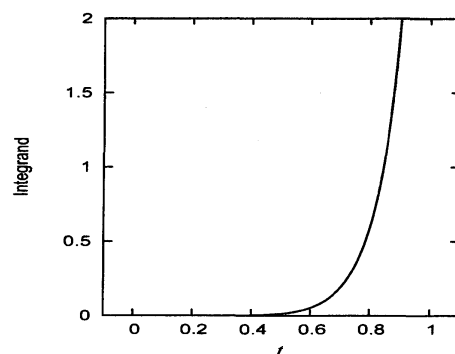
と変形し、極を打ち消してから積分路 C を実軸から遠ざける。 $\sqrt{1+z^6}$ の分岐に注意して、虚軸上の積分に変形して整理すると

$$\begin{aligned} I = & \int_0^\infty \left[\frac{t}{1+t^6 \sinh^2 t} + \Re \frac{2(1+\sqrt{3}i)t}{2-t^6+t^6 \cos((\sqrt{3}+i)t)} \right] dt \\ & + \int_0^1 \frac{t^7}{\sqrt{1-t^6}} \left[\frac{\sinh t \cosh t}{1+t^6 \sinh^2 t} + \Im \frac{(1+\sqrt{3}i) \sin((\sqrt{3}+i)t)}{2-t^6+t^6 \cos((\sqrt{3}+i)t)} \right] dt \end{aligned} \quad (2)$$

となる。(2) 式の被積分関数を図 3 に示す。(2) 式は、 $t \rightarrow \infty$ で $O(t^{-5}e^{-t})$ となる関数の区間 $[0, \infty)$ の積分と、 $t \rightarrow 1$ で $O((1-t)^{-1/2})$ となる関数の区間 $[0, 1)$ の積分の和であり、どちらも積分端点以外は穏やかな関数の積分であり、DE 公式が有効に適用できる。その性能は、 N 桁求めるための関数計算回数は $O(N \log N)$ となる。また、(2) 式の 2 項目の積分は、変数変換 $t = \cos \theta$ で滑らかな周期関数の積分へと変換されるので、直接この周期積分に台形公式を適用するだけで関数計算回数は $O(N)$ とすることも出来る。



第一項の図



第二項の図

図 3: (2) 式の被積分関数

4 超収束する DE 公式

ここでは、(2) 式に特化した DE 公式を作成する。簡略化のために、(2) 式の第 1 項積分を考える。第 1 項積分は、積分内の後半を虚軸上に移し、極 $p \approx 0.3 + 0.9i$ の留数を考慮すると

$$I_1 = - \int_0^\infty \frac{t}{1+t^6 \sinh^2 t} dt + \Im \frac{2\pi p^2}{3 + ip^4 \cos p}$$

となる。次に、 $c > 0$ として

$$I_1 = -\frac{1}{\pi i} \int_{-\infty}^{\infty} \frac{t \log(ict)}{1 + t^6 \sinh^2 t} dt + \Im \frac{2\pi p^2}{3 + ip^4 \cos p}$$

と書き換えて、さらに K を正の整数として、積分路を平行に $-iT$, $T = \pi(K + 1/2)$ だけずらして

$$I_1 = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{(it + T) \log(ict + cT)}{1 + (it + T)^6 \cosh^2 t} dt + R_K$$

と変形する (図 4 参照)。ここで R_K は、 $-ip \approx 0.9 - 0.3i$, $-i\bar{p}$, $-iq \approx 0.4 - 0.9i$, $-i\bar{q}$ および $\pm 1/(\pi k)^3 - \pi ki$ ($k = 1, 2, 3, \dots, K$) 近傍の $2K + 4$ 個の極による留数項である。この積分に変数変換 $t = T' \sinh u$ ($T' = \pi(K - 1/2)$) を施した後、 $c = 1/T$ と置き、刻み幅 h の台形公式を適用すると

$$I_1 = \frac{h}{\pi} \sum_{n=-M}^M \frac{(T + iT' \sinh nh) \log(1 + i(T'/T) \sinh nh) T' \cosh nh}{1 + (T + iT' \sinh nh)^6 \cosh^2(T' \sinh nh)} + R_K + E_{h,2K} + \Delta I_{1,h,K,M}$$

となる。ここで、 $E_{h,2K}$ は $t = -ip, -i\bar{p}, -iq, -i\bar{q}$ および $t \approx \pm 1/(\pi k)^3 - \pi ki$ ($k = 1, 2, 3, \dots, 2K$) 近傍の留数を、誤差の特性関数 [6] を用いて誤差評価することで得られる補正項である。このとき、求める桁数に比例して K を取ることで、近似誤差は $\Delta I_{1,h,K,M} = O(\exp(-C'M)) = O(\exp(-C''(M + 2K)))$ ($C', C'' > 0$) となり、超収束する DE 公式を得る。結果、 N 桁求めるための留数計算を含む関数計算回数は $O(N)$ となる。

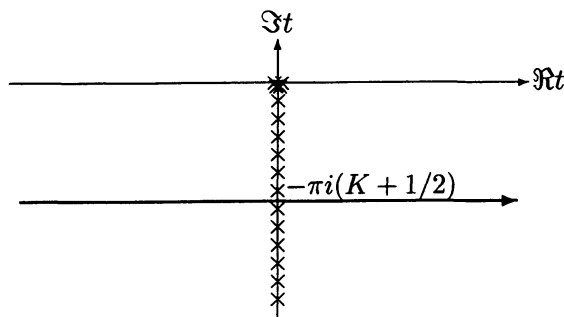


図 4: 積分路

5 一般化された GH 積分

また 3 節で述べた方法は、一般化された GH 積分 $\int_0^{\infty} \frac{x^m}{1 + x^n |\sin x|^l} dx$ において、 l が偶数のときにある条件を満たせば、同様の計算法の導出が可能である。このときの具体的な $\psi(z)$ は

$$\psi(z) = \frac{\pi z^n \sin z \cos z}{\sqrt{1 + z^n}} \frac{z^m}{1 + z^n \sin^2 z} \quad (l = 2),$$

$$\psi(z) = \left(\frac{z^{n/2} \sin^2 z + i}{2} \sqrt{1 - iz^{n/2}} + \frac{z^{n/2} \sin^2 z - i}{2} \sqrt{1 + iz^{n/2}} \right) \\ \cdot \frac{\pi z^{n/2} \sin z \cos z}{\sqrt{1 + z^n}} \frac{z^m}{1 + z^n \sin^4 z} \quad (l = 4), \dots$$

である。さらにその場合、(1) 式から (2) 式のような式の変形が可能な積分については、その収束性も自動的に示されることとなる。

また本稿の変形がうまく適用できないタイプの、より一般の振動積分には、一般化連続オイラー変換 [7] が適用できることがある。その具体的算法は、一般化連続オイラー変換を用いた GH 積分として付録 1 に示す。

6 計算例

四則演算の多倍長計算ライブラリに GMP [8] (ver. 5.0.1) & MPFR [9] (ver. 2.4.2) を用い、その上での初等関数計算法として、ワード単位の CORDIC 法に Taylor 展開を組み合わせた独自の方法を用いて計算を行った。具体的算法は付録 2 に示す。この初等関数計算法は、 N 桁の関数値を得るために必要な乗算回数は $O(\sqrt{N})$ 程度であり、Gauss の算術幾何平均法の $O(\log N)$ と比較してオーダーは上回るが、メモリの量に応じて比例係数を小さくできるというメリットがある。

DE 公式の実装に関して、DE 変換における指数関数の計算を極力排除する算法を用いた。この算法のポイントは、刻み幅を $h = \log 2/K$ (K は整数) に選ぶことと、一番内側のループを刻み幅 Kh ステップで計算することの二つである。このとき、二重指数関数の $n+1$ 番目の値は $a_{n+1} = \exp(\exp((n+1)Kh)) = \exp(2 \exp(nKh)) = a_n^2$ となり、 n 番目の値を二乗するだけで高速に計算できることがわかる。ただし、初期計算に $\log 2$ の計算と、少しの指数関数の計算と配列用のメモリが必要になる。GH 積分の場合、この算法を用いれば、通常の DE 公式の算法と比較して初等関数の計算がおよそ半分に減る。

表 1 に、Opteron (k10) 2.8GHz 4CPU の計算環境での結果を示す。参考のため、100 桁までの積分値を表すと

$I = 1.16965\ 25542\ 24486\ 47772\ 59225\ 81661\ 19775\ 95884\ 81416\ 66271$
 $46180\ 73171\ 51391\ 33835\ 19905\ 81627\ 12111\ 09181\ 62126\ 67625$
 ...

となる。この表より、桁数と関数計算回数はほぼ比例し、1 桁計算するのに約 1.57 回の関数計算が必要になることがわかる。また、計算時間は求める桁数 N のおよそ 2.6 乗程度になることがわかる。これは、関数計算 1 回につき \log , \exp , atan2 , sincos などの初等関数計算が約 2 回と十数回の乗算計算が含まれるためである。

より詳しく計算時間を解析するために、初等関数計算の計算時間と乗算の計算時間との割合を示した値を表 2 に示す。比較のため、MPFR ライブラリ組み込みの初等関数計算時間も示す。この表より、本算法の初等関数計算時間は、求める桁数 N のおよそ 1.6 乗程度になることがわかる。さらに、本算法は MPFR ライブラリのおよそ 10 倍の計算速度であり、数十回分の乗算時間で初等関数を計算していることがわかる。また、GMP ライブラリの乗算時間は桁数 N のおよそ 1.3 乗である。

次に、現在最速の多倍長算法を使った場合の計算時間を推測する。Schönhage-Strassen の乗算アルゴリズムの計算量は、桁数 N に対して $O_B(N \log N \log \log N)$ である。その上で Gauss の算術幾何平均を用いた初等関数計算の計算量は、 $O_B(N(\log N)^2 \log \log N)$ である。このときの GH 積分の計算量は、桁数 N に対して $O_B((N \log N)^2 \log \log N)$ であり、現在実現可能な最速の実行時間もこのオーダーになると推測できる。

表 1: GH 積分の計算例

有効桁数	計算時間 (Real Time)	関数計算回数
10023 桁 (33300bit)	2.80 秒	15738
100243 桁 (333000bit)	845.13 秒 (約 14 分)	157320
1002429 桁 (3330000bit)	476082 秒 (約 5.5 日)	1573177

表 2: \log , \exp , $\operatorname{atan2}$, sincos の平均計算時間 (CPU Time)

有効桁数	本算法	MPFR	乗算時間 T_{mul}
1 万桁 (33300bit)	$16.8T_{\text{mul}}$	$151T_{\text{mul}}$	$T_{\text{mul}} \approx 0.0641\text{m 秒}$
10 万桁 (333000bit)	$25.1T_{\text{mul}}$	$243T_{\text{mul}}$	$T_{\text{mul}} \approx 1.71\text{m 秒}$
100 万桁 (3330000bit)	$65.1T_{\text{mul}}$	$543T_{\text{mul}}$	$T_{\text{mul}} \approx 26.8\text{m 秒}$

7 まとめ

GH 積分を DE 公式の適用可能な積分に置き換え、高速高精度計算のための様々な改良を行い、100 万桁以上の高精度で計算を行った。以下に、本研究で得られた結果を示す。

1. 極が原因で計算困難な積分に対して、Hilbert 変換などにより共役な被積分関数をうまく見つけることで、いくつかの計算を容易かつ可能にした。
2. ある種の積分 (被積分関数の極が規則的に並び、極の留数計算が容易である積分など) に関して超収束する DE 公式を作成し、 N 桁の積分値を得るための関数計算回数を $O(N)$ にすることを可能にした。
3. 初等関数計算の高速化を試み、既存の高速ライブラリである MPFR のおよそ 10 倍の計算速度を達成した。
4. 指数関数計算を極力排除した DE 公式の実装することで、積分の計算速度を向上させた。特に、GH 積分の計算においては、計算速度をおよそ倍にした。この算法は、すべてのタイプの DE 公式に適用可能で、特に多倍長計算で有効である。

付録 1 一般化連続オイラー変換による GH 積分の計算

複雑な振動積分に対する計算方法として、一般化連続オイラー変換 [7] を用いる方法を紹介する。積分

$$I = \int_0^{\infty} f(x) dx$$

に対する一般化連続オイラー変換は、

$$I_L = \int_0^L f(x) w_L(x) dx, \quad L > 0$$

であらわされる。このとき、 f の性質に応じて w_L を巧妙に選ぶことで、比較的小さい L に関する I_L の値を I の高精度近似とすることができる。具体的な重み w_L は、エルミート関数を

$$h_n(x) = (-1)^n \frac{d^n}{dx^n} e^{-x^2/2}, \quad h_{-1}(x) = \int_x^{\infty} e^{-t^2/2} dt = \sqrt{\frac{\pi}{2}} \operatorname{erfc}(x/\sqrt{2})$$

としたとき、

$$w_L(x) = \sum_{n=0}^M \frac{2^n (x + \alpha)^n}{\sqrt{2\pi n!} (\sigma^2 L)^{n/2}} h_{n-1}((2x - L)/(\sigma L^{1/2})), \quad \alpha, \sigma > 0$$

で与えられる。この変換は、重みをかけて有限区間で打ち切るという単純な操作であるので、様々な振動積分に対して有効であり、 L を大きくしたとき、 M などのパラメータを L に依存させて適切に選べば、 L にほぼ比例する桁数で I_L と I を一致させることができる。したがって、収束の遅い振動積分を指数関数で収束する積分とみなすことができる。ただし、級数の加速法と同様で、 f の振動周期や収束のオーダーなどの性質を前もって把握しておいた上で、その情報をパラメータなどで w_L に反映させなければ I_L の高精度近似は期待できないことに注意する。

この変換を用いた、GH 積分 (1) の具体的な計算方法を示す。まず、

$$g(x) = \sum_{k=0}^{\lfloor L/\pi \rfloor} f(x + \pi k) w_L(x + \pi k), \quad f(x) = \frac{x}{1 + x^6 \sin^2 x}$$

とおく。次に、DE 公式で $g(x)$ を区間 $[0, \pi]$ で積分する。 w_L のパラメータの選び方は、求める桁数を N とするとき、半理論的と半経験的に

$$L = 9N, \quad M = \lfloor 3N/(2 + \log N) \rfloor, \quad \sigma = 1, \quad \alpha = 1/2$$

とすれば、うまく計算できる。ただし、 M を大きくすると $w_L(x)$ が大きな正と負の値で振動するようになるので、桁落ちに注意する必要がある。この方法で、 N 桁の値を得るための関数 f の計算回数は $O((N \log N)^2)$ となる。

付録 2 多倍長初等関数の高速算法

R ビット単位の K 段 CORDIC 法に Taylor 展開を組み合わせた算法の具体例を示す。まず、指数対数関数の計算を示す。

初期計算:

$L(j, k) := \log(1 + j/2^{Rk})$ ($j = 1, 2, \dots, 2^R - 1, k = 1, 2, \dots, K$) を計算しておく。

指数関数 $y = \exp x$ ($0 \leq x < \log 2$) の算法:

```

for  $k = 1, 2, \dots, K$ , do
   $m_k := 0$ 
  for  $r = R - 1, R - 2, \dots, 0$ , do
     $j := m_k + 2^r$ 
    if  $x \geq L(j, k)$  then  $m_k := j$ 
  end for
  if  $m_k > 0$  then  $x := x - L(m_k, k)$ 
end for
 $y := \exp x$  ( $0 \leq x < 2^{-RK}$  であるのを考慮して, Taylor 展開で計算する)
for  $k = 1, 2, \dots, K$ , do
  if  $m_k > 0$  then  $y := y + y * m_k / 2^{Rk}$ 
end for

```

ここで, $y * m_k / 2^{Rk}$ の計算は, ビット演算や整数乗算を組み合わせる高速化する。

対数関数 $y = \log x$ ($1/2 < x \leq 1$) の算法:

```

 $t := 1 - x, y := 0$ 
for  $k = 1, 2, \dots, K$ , do
   $x := 1 - t, m := 0$ 
  for  $r = R - 1, R - 2, \dots, 0$ , do
    if  $x/2^{Rk-r} \leq t$  then  $t := t - x/2^{Rk-r}, m := m + 2^r$ 
  end for
  if  $m > 0$  then  $y := y - L(m, k)$ 
end for
 $y := y + \log(1 - t)$  (Taylor 展開で計算する)

```

ここで, 一番内側のループを高速化するため, 境界の値以外は低い精度で行うという工夫をする。さらに, \log の Taylor 展開を高速化するため, $\log(1-t) = -2 \tanh^{-1}(t/(2-t))$, $\tanh^{-1} x = x + x^3/3 + x^5/5 + x^7/7 + \dots$ と展開して計算する。

次に, 三角逆三角関数の計算を示す。

初期計算:

$T(j, k) := \tan^{-1}(j/2^{Rk})$ ($j = 1, 2, \dots, 2^R - 1, k = 1, 2, \dots, K$) を計算しておく.

三角関数 $y = \sin x, z = \cos x$ ($0 \leq x < \pi/4$) の算法:

```

 $x := x/2$  (半角)
for  $k = 1, 2, \dots, K$ , do
   $m_k := 0$ 
  for  $r = R - 1, R - 2, \dots, 0$ , do
     $j := m_k + 2^r$ 
    if  $x \geq T(j, k)$  then  $m_k := j$ 
  end for
  if  $m_k > 0$  then  $x := x - T(m_k, k)$ 
end for
 $y := \sin x, z := \cos x$  (Taylor 展開で計算する)
for  $k = 1, 2, \dots, K$ , do
  if  $m_k > 0$  then
     $t := z * m_k / 2^{Rk}, u := y * m_k / 2^{Rk}$ 
     $y := y + t, z := z - u$ 
  end if
end for
 $t := 2 * y / (y^2 + z^2), u := y$ 
 $y := z * t, z := 1 - u * t$  (倍角 & 正規化)

```

逆三角関数 $z = \text{atan2}(x, y) = \tan^{-1}(x/y)$ ($0 \leq \arg(y + ix) < \pi/4$) の算法:

```

 $z := 0$ 
for  $k = 1, 2, \dots, K$ , do
   $t := x, m := 0$ 
  for  $r = R - 1, R - 2, \dots, 0$ , do
    if  $y/2^{Rk-r} \leq t$  then  $t := t - y/2^{Rk-r}, m := m + 2^r$ 
  end for
  if  $m > 0$  then
     $t := y * m / 2^{Rk}, u := x * m / 2^{Rk}$ 
     $x := x - t, y := y + u$ 
     $z := z + T(m, k)$ 
  end if
end for
 $z := z + \tan^{-1}(x/y)$  (Taylor 展開で計算する)

```

最後に、多倍長 Taylor 展開のちょっとした高速化技法を示す。簡略化のために、 $y = \sum_{n=0}^M x^n/a_n$ という形の和の計算で、 a_n が整数と仮定する。そして、 $M < M'2^P$ となる

適当な正の整数 M', P を求めておく.

高速多倍長 Taylor 展開の算法:

```

for  $k = 0, 1, \dots, 2^P - 1$ , do
   $s_k := 1/a_k$ 
end for
 $t := 1, u := x^{2^P}, y := 0$ 
for  $m = 1, 2, \dots, M' - 1$ , do
   $t := t * u$ 
  for  $k = 0, 1, \dots, 2^P - 1$ , do
     $s_k := s_k + t/a_{k+m2^P}$ 
  end for
end for
for  $k = 2^P - 1, 2^P - 2, \dots, 0$ , do
   $y := y * x + s_k$ 
end for

```

この算法には, $M' + 2^P + P - 3$ 回の乗算と $M'2^P$ 回の整数除算と和の演算が含まれる. したがって, $M' \approx \sqrt{M}$ 程度に選べば, 乗算回数は $O(\sqrt{M})$ 程度に減る.

参考文献

- [1] 高木貞治, 解析概論 改訂第3版, 岩波書店, (1983).
- [2] E. Goursat, Cours D'analyse Mathematique Tome I, Gauthier Villars, Paris, (1902).
- [3] G. H. Hardy, Messenger of Mathematics Vol. 31, (1902).
- [4] 二宮市三, 加速法による数値積分の計算の一例, 京都大学数理解析研究所講究録 585, (1986).
- [5] Y. Hatano, I. Ninomiya, H. Sugiura, T. Hasegawa, Numerical evaluation of Goursat's infinite integral, Numerical Algorithms Vol. 52, (2009).
- [6] H. Takahasi, M. Mori, Double exponential formulas for numerical integration, Publ. Res. Inst. Math. Sci. 9, (1974).
- [7] T. Ooura, A generalization of the continuous Euler transformation and its application to numerical quadrature, Journal of Computational and Applied Mathematics, Vol. 157, (2003).
- [8] GNU Multiple Precision Arithmetic Library <http://gmplib.org/>
- [9] GNU MPFR Library <http://www.mpfr.org/>